COMPLETE SET OF PENDING CLAIMS

1. (Currently amended) A processor for reading instructions from a memory according to a program counter, and for executing the read instruction,

the memory storing, in a position corresponding to a byte boundary, at least one processing packet being made of an integer number of bytes, the processing packet including processing target instructions, each processing target instruction being an operation to be executed by the processor, the number of processing target instructions being any number except for a power of 2,

the program counter including a first program counter and a second program counter,

the first program counter indicating a storage position of ~~a~~ the processing packet in the memory, ~~the processing packet being made of an integer number of bytes, the storage position being a position corresponding to a byte boundary,~~

the second program counter indicating a position of processing target instruction in the processing packet by using the same number of values as the number of processing target instructions, and cycling through the values, ~~the processing target instruction being an operation to be executed by the processor, and the position of which does not correspond to a byte boundary~~.

2. (Original) The processor of Claim 1, including a first program counter updating means and a second program counter updating means,

the second program counter updating means incrementing a value of the second program counter in accordance with an amount of instructions that were executed in a preceding cycle and sending any carry generated in an incrementing to the first program counter updating means, and

the first program counter updating means adding the carry received from the second program counter updating means to the value of the first program counter.

3. (Original) The processor of Claim 2, further including:

program counter relative value extracting means for extracting, when an instruction being executed includes a program counter relative value that is based on an address of a first instruction executed in a present cycle, the program counter relative value; and

calculating means for adding the program counter relative value to the value of the first program counter and the value of the second program counter, and setting an addition result as the value of the first program counter and the value of the second program counter.

4. (Original) The processor of Claim 3,

wherein the calculating means includes a first calculating unit and a second calculating unit,

the second calculating unit adding the value of the second program counter and lower bits of the program counter relative value, setting a result of an addition as the value of the second program counter, and sending any carry generated in the addition to the first calculating unit,

the first calculating unit adding the value of the first program counter, upper bits of the program counter relative value, and any carry received from the second calculating unit, and setting a result of an addition as the value of the first program counter.

5. (Original) The processor of Claim 3,

wherein the calculating means includes a first calculating unit and a second calculating

unit,

the second calculating unit adding the value of the second program counter and lower bits of the program counter relative value without generating a carry, and setting a result of an addition as the value of the second program counter,

the first calculating unit adding the value of the first program counter and upper bits of the program counter relative value, and setting a result of an addition as the value of the first program counter.

6. (Original) The program counter of Claim 3,

wherein the calculating means adds the value of the first program counter and upper bits of the program counter relative value, sets a result of an addition as the value of the first program counter, and sets lower bits of the program counter relative value as the value of the second program counter.

7. (Original) The processor of Claim 3,

wherein the calculating means adds the program counter relative value and a value whose upper bits are the value of the first program counter and lower bits are the value of the second program counter, and sets upper bits of a result of an addition as the value of the first program counter and lower bits of the result as the second program counter.

8. (Original) The processor of Claim 2, further including:

program counter relative value extracting means for extracting, when an executed instruction includes a program counter relative value that is based on an address of the executed

instruction, the program counter relative value;

program counter amending means for amending the value of the first program counter and the value of the second program counter to indicate an address of the executed instruction; and

calculating means for adding the program counter relative value, the value of the first program counter, and the value of the second program counter, and setting a result of an addition as the value of the first program counter and the value of the second program counter.

9. (Original) The processor of Claim 2, further including:

program counter relative value calculating instruction decoding means for decoding a program counter relative value calculating instruction that performs an addition using a program counter relative value and one of

(a) a value of the program counter stored in a register, and

(b) the value of the first program counter and the value of the second program counter;

calculating means for performing the addition indicated by the program counter relative value calculating instruction to generate an addition result; and

program counter value updating means for storing the addition result in one of

(a) the register, and

(b) the first program counter and the second program counter.

10. (Previously amended) The processor of Claim 1,

wherein the first program counter indicates bits of a memory address more significant than a $1 + log_2 n^{th}$ bit from a least significant, the memory address specifying the storage position

of the processing packet in memory, and *n* being a length of a processing packet in bytes.

11. (Previously amended) The processor of Claim 10, further including

an instruction buffer for temporarily storing instructions; and

instruction reading means for transferring instructions being made of an integer number of bytes from the memory to the instruction buffer, in accordance with available space in the instruction buffer but regardless of a size of a processing packet.

49. (Currently amended) A processor for reading instructions from a memory according to a program counter, and for executing the read instruction,

the memory storing, in a position corresponding to a byte boundary, at least one processing packet being made of an integer number of bytes, the processing packet including processing target instructions, each processing target instruction being an operation to be executed by the processor, the number of processing target instructions being any number except for a power of 2,

the program counter including a first program counter and a second program counter,

the first program counter indicating a storage position of ~~a~~ the processing packet in the memory, ~~the processing packet being made of an integer number of bytes, the storage position being a position corresponding to a byte boundary,~~

the second program counter indicating a position of processing target instruction in the processing packet by using the same number of values as the number of processing target instructions, and cycling through the values, ~~by cycling through m different values, with m not being a power of 2,~~ and sending a carry to the first program counter if the second program

counter cycles~~, the processing target instruction being an operation to be executed by the processor, and the position of which does not correspond to a byte boundary~~.

50. (New) A processor system for reading and executing a plurality of target instructions comprising:

a memory for storing a packet which includes the plurality of target instructions where each of the plurality of target instructions is an operation executable by the processor, the number of the plurality of target instructions is equal to m where m is a number other than a number equal to 2^n, where n is a positive integer greater than or equal to 1;

a first program counter, operably coupled to the memory, for identifying the location of the packet in the memory; and

a second program counter, operably coupled to the memory, for identifying the location of each of the plurality of instructions in the packet by using the number of the plurality of target instructions to proceed to a preceding or subsequent target instruction.